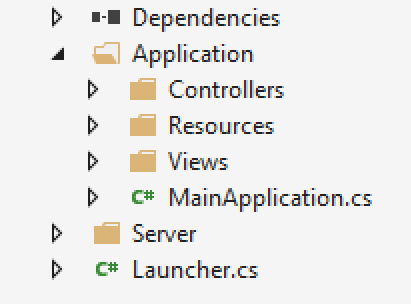
# Exercise: Web Server – HTTP Protocol

Problems for exercises and homework for the [“C# Web Basics” course @ SoftUni](https://softuni.bg/courses/csharp-web-development-basics). Yoy can submit your solution in the course web page.

**Before you begin make sure you have implemented the Web Server from the previous lab.**

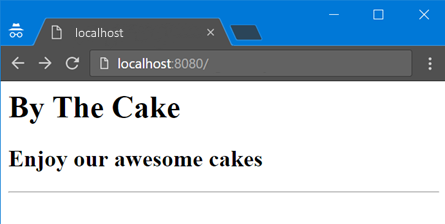
You can use similar file structure for your applications:



## First Web Site – By The Cake App

Create application called **ByTheCake**. Create a main page that contains **h1** header **“By the Cake”** and **h2** header **“Enjoy our awesome cakes”**. Put a **horizontal line** afterwards. The path to the page should be **"localhost:{port}/".**

### Example

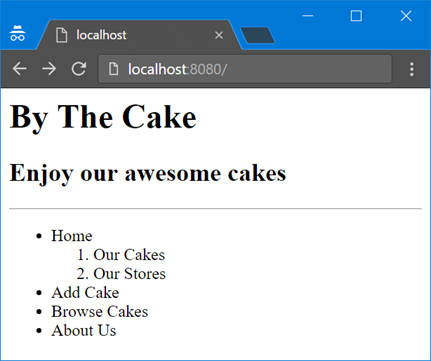


Hint: Create file index.html and place it in the Resources folder. You can read the html from the index file and return it as a response.

## By the Cake: Add Menu

Extend your main page. **Add a menu** as an **unordered list**. It should have four menu items. The Home item should have an ordered sub list with two list items. See the example below.

### Example



## By the Cake: Add Paragraph

Extend your web page. Add **h2** header called **“Home”**. It should have **two sections**. Each section has a **h3 header**, a **paragraph** and **an image**.

**Section one** should be with header **“Our cakes”**, paragraph with the following **text:**

*Cake is a form of sweet dessert that is typically baked. In its oldest forms, cakes were modifications of breads, but cakes now cover a wide range of preparations that can be simple or elaborate, and that share features with other desserts such as pastries, meringues, custards, and pies.*

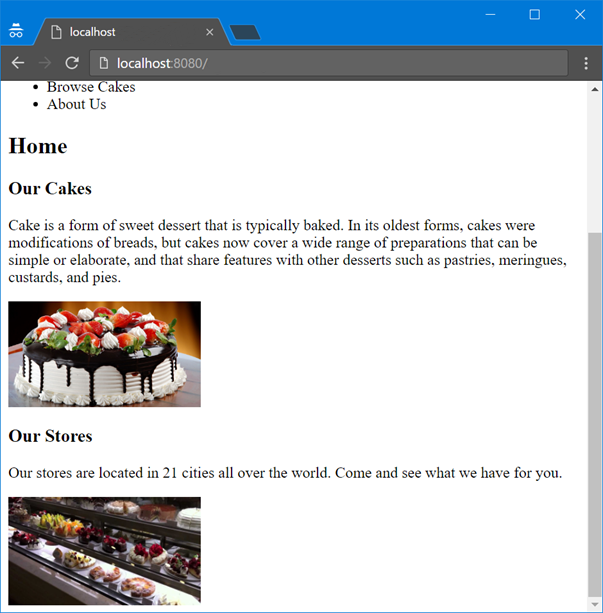
and a **random cake image**.

**Section two** should be with **header “Our Stores”**, paragraph with the following **text**:

*Our stores are located in 21 cities all over the world. Come and see what we have for you.*

and a **random cake store image**.

### Example

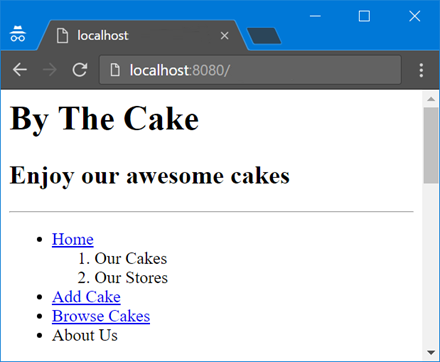


## By the Cake: Add Links

Extend your web page. **Add links for the menu we have**.

* **Home** item should reference **your current page "localhost:{port}/ "**.
* **Add Cake** should reference to "**localhost:{port}/add"**
* **Browse Cakes** should reference to **" localhost:{port}/search"**.

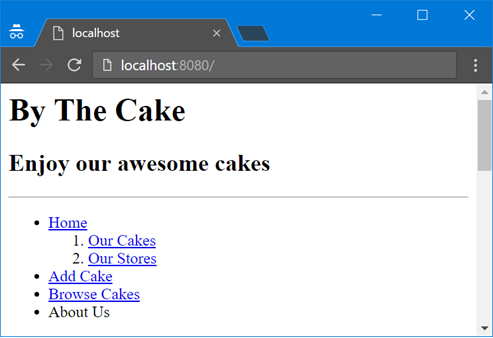
### Example

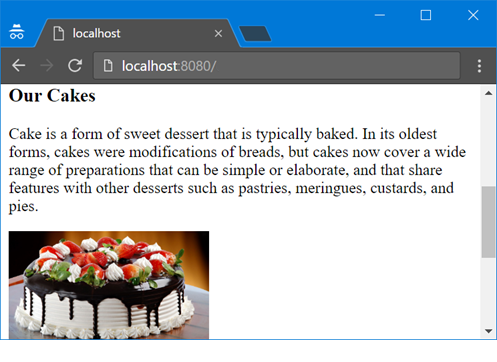


## By the Cake: Add Anchors

Extend your web page. Add an **id attribute** to **h3** **Our cakes** with value “**cakes**”. Add another id attribute to **h3** **Our stores** with value “**stores**”. The menu item “**Our cakes**” should reference to “**cakes”**. The menu item **Our stores** should reference “**stores”**.

### Example

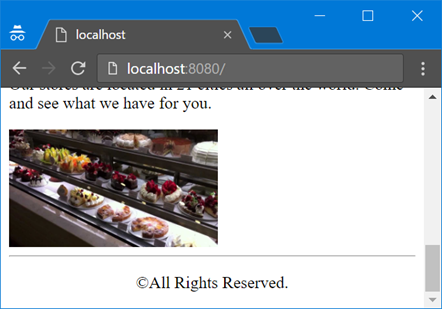




## By the Cake: Add Footer

Extend your web page. Add **footer** which says **©All Rights Reserved**. There should be a **horizontal line above the footer**. **Center the text**.

### Example



### Hint

* Use the **<footer>** tag
* To center the text of your paragraph **use CSS**. Consider the **text-align**property.

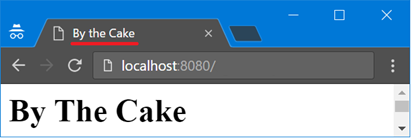
## By the Cake: Add Head

Extend your web page. Now it is time to **add some meta data**. We need to set the following properties:

* **Title**: By the Cake
* **Charset**: UTF-8
* **Description**: Buy the cake in By the Cake
* **Keywords**: cakes, buy
* **Author**: Your name

The meta data will help search engines to recognize your web site as a best match.

### Examples



### Hint

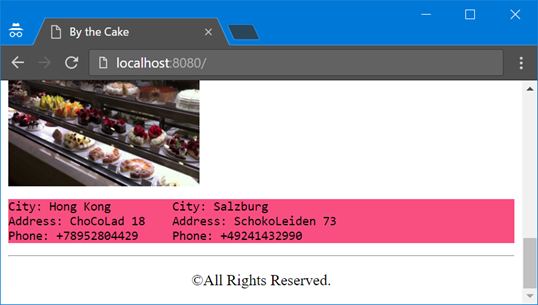
Use the **<head>** tag

## By the Cake: Add Stores Info

Extend your web page. Add **preformatted text above the footer** and below the last picture. Change the **background color** to **#F94F80**. The text should contain information about the stores as follows:

|  |  |
| --- | --- |
| City: Hong Kong Address: ChoCoLad 18 Phone: +78952804429 | City: Salzburg Address: SchokoLeiden 73 Phone: +49241432990 |

### Example



### Hint

* Use **<pre>** tag
* Use CSS only to change the background color

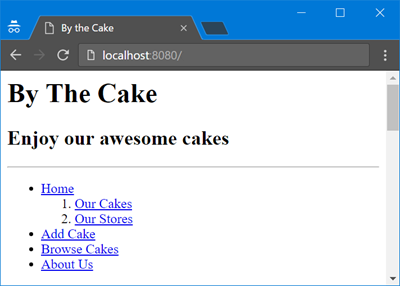
## By the Cake: About Us Page

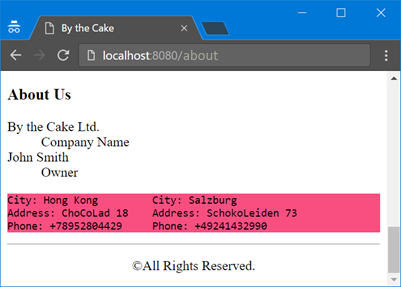
Extend your web site. Create **About Us page** with **h2 header** with text “**About us**” and path "**localhost:{port}/about**" followed by a **definition list**. Put the information just above the store info. The definition list should have two texts with a single definition each.

* By the Cake Ltd.
  + Company Name
* {Your Name}
  + Owner

Create an anchor that will lead to “About us” just like in Exercise 5.

### Examples

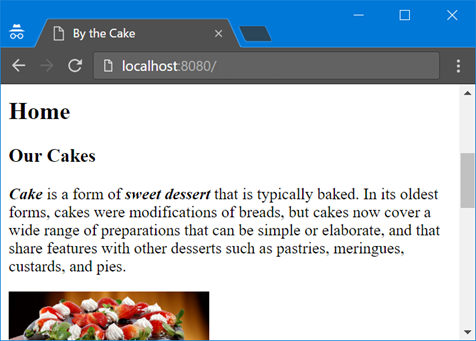




## By the Cake: Emphasize the Words

Extend yourmain page. Find the words **sweet desert, cake, store** in all paragraphs and make them **bold** and *italic*.

### Example

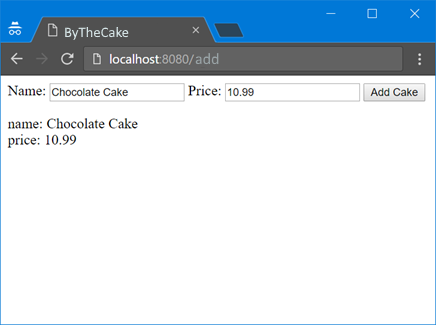


## By the Cake: Add Page

Extend your web site. Create a page which will be responsible for adding cakes functionality. The page should have path "**localhost:{port}/add**". It should have a simple form to add cakes. **Create a Cake class**. Each cake should have name and price. When you click the submit button a new cake should be created and saved in a list. The newly created cake should be printed below the form. **“name”** and **“price”** are the parameters of the request.

**In order for the form to work create a functionality which processes POST requests.**

### Example

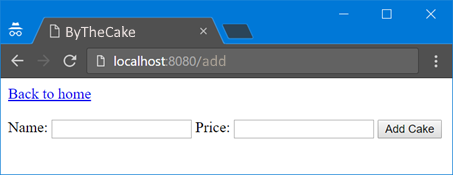


### Hint

Use the method attribute of the form to pass a POST request and the property FormData of the HttpRequest object to obtain parameters.

## By the Cake: Write data

Extend your web application. The submitted data should be appended on a new row split by comma in a file called **database.csv**. Add a link to go back to your main page ("localhost:{port}/").



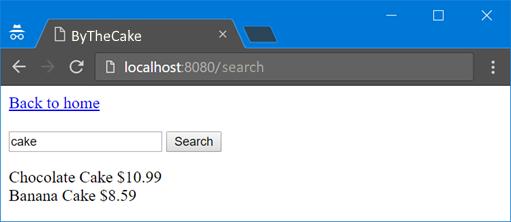
## By the Cake: Search Cakes

Extend your web site. Crate a new page that contain a single form **with GET method**. The path to the page should be "**localhost:{port}/search**" See the example below. It will be used to **search cakes by name**. Cakes should be searched from **database.csv** and printed below the form.

Add a functionality to go back to your main page (**"localhost:{port}/"**).

### Example

|  |
| --- |
| **database.csv** |
| Chocolate cake,10.99 Banana cake,8.59  Strawberry Cak, 10.54 |

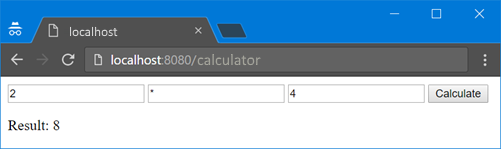
****

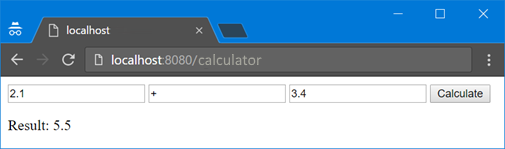
## Calculator App

Write an application that prints a form with three inputs on the browser followed by submit button. Input 1 and input 3 should receive numbers. Input 2 should receive a **mathematical sign**. When you press the button the result of the equation should appear. See the example below. There are four possible operations: **add, subtract, multiply and divide (+, -, \*, /)**. If none of the above is entered print an error **Invalid Sign**.

**Use POST method.**

### Examples

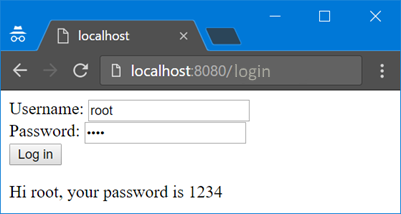




## Login Form

Write a C# application that prints a form with two inputs on the browser followed by а submit button. Inputs should receive username and password. When you click the button, a message saying “Hi {username}, your password is {password}” should appear. Use **POST** method**.**

### Example



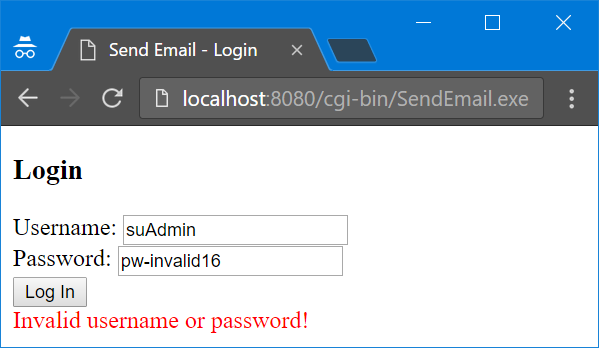
## \*Send Email

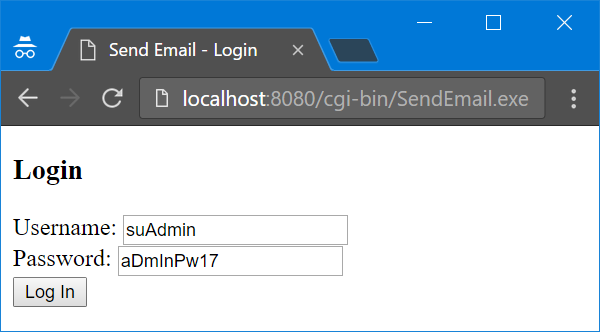
Extend the previous application so that with functionality for sending email. Add a page that contains a form with 2 text fields for **username** and **password**. A user must be able to log in only if the **provided username** is suAdmin and the **password** is aDmInPw17. When is logged in he/she should be able to **send email via provided form** where recipient email, subject and body of the email must be provided. Before sending an email validate the provided data:

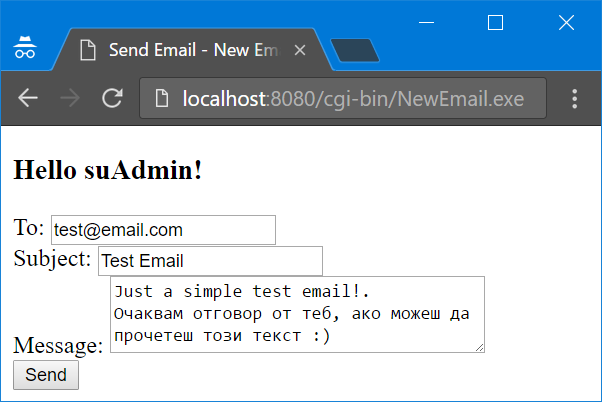
* **To** - must be a **valid email** (search in the internet for the requirements of an email address)
* **Subject** - must be no longer than **100 symbols**
* **Message** - must be able to send **all Unicode characters**

The email messages must be sent from **your own email address**.

### Example





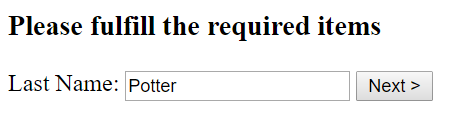


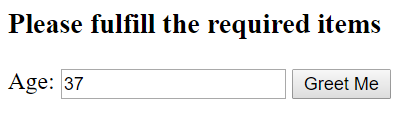
## \*Greetings

Write a program that collects some data from the user and finally prints a greeting message. Check the examples below.

### Example









## \*\*Survey app

Write a C# application that creates a **survey for receiving feedback from users for a given product**. Look at the example to check what fields the feedback form should have and how they should be styled. In the **Status** field, the possible options are:

* Student
* Part-time Employee
* Full-time Employee
* Unemployed
* Do not want to answer

Make **appropriate validations** where it is necessary.

When the form is submitted **store each submission** in a file called **survey-results.csv**

### Example

